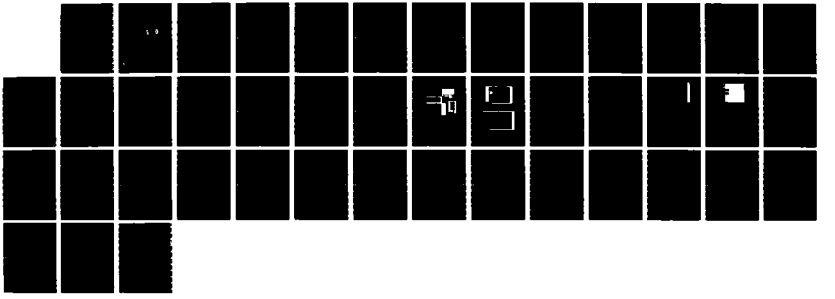
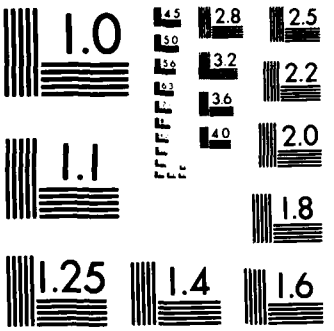
AD-A171 795 MASTIF (MIT ANALYSIS AND SYNTHESIS TOOL FOR IC FABRICATION) - A WORKSTATI..(U) MASSACHUSETTS INST OF TECH CAMBRIDGE DEPT OF ELECTRICAL ENGIN.. D S BONING

1/1

UNCLASSIFIED JUN 86 VLSI-MEMO-86-325 N00014-85-K-0213 F/G 9/5 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A171 795

VLSI Memo No. 86-325
June 1986

Contract N00014-85-K-0213

MASTIF - A WORKSTATION APPROACH TO INTEGRATED CIRCUIT PROCESS AND DEVICE DESIGN

Duane S. Boning

DTIC
ELECTE
SEP 09 1986
S D

Abstract

The design of integrated circuits increasingly requires the use of computer-aided design tools. While collections of integrated tools exist for system, logic, circuit, and layout aspects of VLSI design, very few similar systems have been built to address process and device design. The research reported in this thesis has had three goals. First, the requirements of a process and device design environment have been investigated. It is proposed that functional capabilities in specification, synthesis, capture, verification, simulation, and analysis are all needed in both the process design and device design domains. The second goal of this research has been to implement a subset of these functions. While simulation programs already exist, limited capabilities in capture, synthesis, verification, and analysis have been implemented as part of this thesis. The third goal has been to integrate these tools into a user-friendly design environment. A workstation providing graphic, window-oriented user-interaction has been implemented to satisfy this third goal. The MASTIF (MIT Analysis and Synthesis Tool for IC Fabrication) workstation is described and illustrated in this thesis.

DTIC FILE COPY

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

86 8 8 040

Microsystems Research Center
Room 39-321
Massachusetts Institute of Technology
Cambridge
Massachusetts
02139

Acknowledgements

Submitted to the Department of Electrical Engineering and Computer Science, MIT, May 16, 1986 in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering and Computer Science. This work was supported in part by the Defense Advanced Research Projects Agency under contract no. N00014-85-K-0213 and by a National Science Foundation Graduate Fellowship.

Author Information

Boning: Department of Electrical Engineering and Computer Science, MIT, Room 39-315, Cambridge, MA 02139, (617) 253-0450.

Copyright (c) 1986, MIT. Memos in this series are for use inside MIT and are not considered to be published merely by virtue of appearing in this series. This copy is for private circulation only and may not be further copied or distributed, except for government purposes, if the paper acknowledges U. S. Government sponsorship. References to this work should be either to the published version, if any, or in the form "private communication." For information about the ideas expressed herein, contact the author directly. For information about this series, contact Microsystems Research Center, Room 39-321, MIT, Cambridge, MA 02139; (617) 253-8138.

MASTIF A WORKSTATION APPROACH TO INTEGRATED CIRCUIT PROCESS AND DEVICE DESIGN

by

Duane S. Boning

B.S., Massachusetts Institute of Technology

(1984)

submitted in partial fulfillment

of the requirements for the degree of

MASTER OF SCIENCE

IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 16, 1986

©Massachusetts Institute of Technology 1986

Signature of Author Duane S Boning

Department of Electrical Engineering and Computer Science

May 16, 1986

Certified by Dim. A. Antoniadis

Dimitri A. Antoniadis

Thesis Supervisor

Accepted by _____

Arthur C. Smith

Chairman, Departmental Committee on Graduate Students

MASTIF – A Workstation Approach to Integrated Circuit Process and Device Design

by

Duane S. Boning

Submitted to the Department of Electrical Engineering
and Computer Science on May 16, 1986 in partial
fulfillment of the requirements for the Degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

The design of integrated circuits increasingly requires the use of computer aided design tools. While collections of integrated tools exist for system, logic, circuit, and layout aspects of VLSI design, very few similar systems have been built to address process and device design. The research reported in this thesis has had three goals. First, the requirements of a process and device design environment have been investigated. It is proposed that functional capabilities in specification, synthesis, capture, verification, simulation, and analysis are all needed in both the process design and device design domains. The second goal of this research has been to implement a subset of these functions. While simulation programs already exist, limited capabilities in capture, synthesis, verification, and analysis have been implemented as part of this thesis. The third goal has been to integrate these tools into a user-friendly design environment. A workstation providing graphic, window-oriented user-interaction has been implemented to satisfy this third goal. The MASTIF (MIT Analysis and Synthesis Tool for IC Fabrication) workstation is described and illustrated in this thesis.

Thesis Supervisor: Dimitri A. Antoniadis
Title: Professor of Electrical Engineering and Computer Science

Acknowledgements

This work has been performed in the MIT Microsystems Technology Laboratory as part of the Computer Aided Fabrication (CAF) research project. Support for this research has been provided in part by DARPA contract N00014-85K-0213 and by the National Science Foundation through a Graduate Fellowship. I greatly appreciate the guidance of my research advisor, Dimitri Antoniadis; he has provided the environment and imagination of purpose that have made MASTIF possible. Several people have contributed to the development of MASTIF, both with working code and through valuable discussions. These people are Jarvis Jacobs, Thye-Lai Tung, Robert Harris, Rex Lowther, Ron Duncan, and Dave Hamilton. In addition to financial and intellectual support, I also acknowledge and appreciate the considerable emotional support provided by my fiancée Margaret Norris.

Accesion For
NTIS CRA&I
DTIC TAB
Unannounced
Justification
By ltr. on file
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

Biography

Duane Boning earned the Bachelor of Science in Electrical Engineering and the Bachelor of Science in Computer Science degrees in 1984 at the Massachusetts Institute of Technology. He is a member of Tau Beta Pi, Etta Kappa Nu, and Sigma Xi, and is a student member of the IEEE and the ACM. He has been a General Motors Scholar, and is currently a National Science Foundation Graduate Fellow. His primary research interest is in the computer aided design of fabrication processes and devices, and in the computer aided fabrication of integrated circuits.

Contents

Abstract 1

Acknowledgements 2

Biography 3

Contents 4

1 Introduction 7

2 Requirements for a Process and Device Design System 10

2.1 Process Specification 10

2.2 Process Synthesis 10

2.2.1 Automated Synthesis 11

2.2.2 Synthesis by the Designer 11

2.3 Process Capture 11

2.4 Process Verification 11

2.5 Process Simulation 12

2.6 Process Analysis 13

2.7 Device Specification 13

2.8 Device Synthesis 13

2.9 Device Capture 13

2.10 Device Verification 14

2.11 Device Simulation 14

2.12 Device Analysis 14

3 MASTIF Capabilities 16

3.1 Process Description Window 16

3.2 Cross Section Summary Window 19

3.3 Process Simulation Windows 19

3.4 SUPREM-III Plot Window 21

3.5 Device Simulation Windows 22

3.6 MIDAS Window 22

3.7 Other Windows 23

4 MASTIF Implementation **24**

4.1 Hardware and Imbedded Software 24

4.2 MASTIF Information Storage 24

4.3 MASTIF Software 25

4.4 Window Management Subsystem 26

4.4.1 MASTIF Windows 26

4.4.2 MASTIF Menus 26

4.4.3 Input/Output Handler 26

4.5 Syntax and Parsing Subsystem 26

4.5.1 Input File Format 27

4.5.2 Parameter Specification Grammar 27

4.5.3 Interactive Input File Manipulator 29

4.6 Subprocess Handling Subsystem 29

5 Results **30**

5.1 Model of Process and Device Engineering 30

5.2 Workstation Approach 32

5.3 Integration of Process and Device Tools 33

6 Future Work **34**

6.1 Profile Interchange Format 34

6.2 Process Description Language 34

References **36**

Appendix - MASTIF User's Manual **40**

List of Figures

1 VLSI design hierarchy and functional design aids. 7
2 MASTIF screen. 17
3 MASTIF *Process Description Windows*. 18
4 Defining one-dimensional cross sections. 20
5 MASTIF *Cross Section Summary Window*. 21
6 SUPREM-III *Process Simulation Window*. 22
7 SUPREM-III *Plot Window* 23
8 MASTIF information storage. 25
9 Parameter Specification Syntax. 28

List of Tables

1 MASTIF *Process Description* Statements. 19

1 Introduction

Spurred by the increasing complexity of integrated circuits, many CAD tools have been developed to provide help in successive phases of IC design. These tools span the various levels of VLSI design, including architecture, system, logic, circuit, device, and process levels [1]. This design hierarchy is shown in Figure 1. Design in several of these domains can be accomplished on "integrated workstations," where tools for specification, capture, and synthesis are accompanied by verification, simulation, and analysis programs [2].

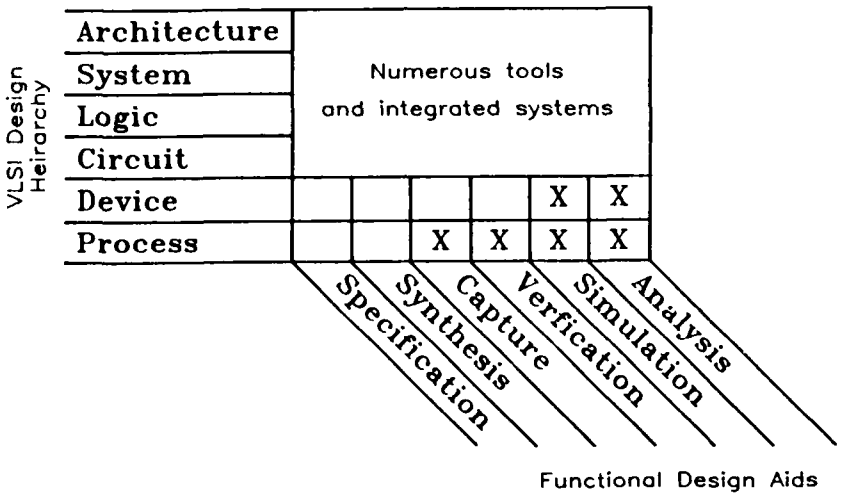


Figure 1: VLSI design hierarchy and functional design aids.
The functions marked by (X) are included in the MASTIF workstation.

The design of semiconductor devices and fabrication processes, on the other hand, is a task which suffers from the lack of a similar workstation approach. A limited number of tools do provide simulation capability: process simulators such as SUPREM-III [3] and SUPRA [4] have been developed to model aspects of the fabrication process; device simulators such as MINIMOS [5] also exist for the evaluation of MOS and other semiconductor devices. Nevertheless, process and device design has long been the poor cousin in the VLSI computer aided design family; as of yet there are very few integrated systems which provide a full complement of tools (that is, any tools beyond simulation) in the domain of device and process design [6]. This introduction will show that there is a natural evolution of CAD

tools in all areas of VLSI design, and that a parallel evolution of tools for process and device design is to be expected in the future.

This thesis has a twofold purpose. If process and device tools are to evolve in the same way as other VLSI CAD tools, then by analogy the same types of abstract capabilities will be needed. In Section 2 of this thesis, I will examine the requirements and composition of a complete process and device design system. In a sense, Section 2 will serve as a blueprint for the future evolution of process and device design tools.

The second purpose of this thesis is to describe the MASTIF (MIT Analysis and Synthesis Tool for IC Fabrication) workstation. Implemented as part of this research, this workstation meets some of the needs beyond simulation for process and device design. The capabilities of MASTIF are discussed in Section 3, and the implementation of the workstation described in Section 4. Results of this research are summarized in Section 5, and ongoing and future work is presented in Section 6.

A set of MASTIF documentation has been written, part of which is included in this thesis. The MASTIF **User's Manual** contains more detailed information about the use of the workstation, and is included in here as an appendix. The MASTIF **Programmer's Manual** contains a detailed description of MASTIF modules and implementation issues and is intended to aid in the future development of MASTIF. Finally, the MASTIF **Manager's Manual** provides documentation on the installation of MASTIF, as well as a description of common problems.

Evolution of VLSI CAD Tools

The evolution of CAD (Computer-Aided-Design) tools in various domains of VLSI design follows a recurrent pattern. Under the pressures of increasing model size and complexity, simple hand and paper methods grow into sophisticated CAD tools with specification, synthesis, capture, verification, simulation, and analysis aspects, and finally achieve integration with other CAD tools [1].

This evolution is illustrated by the growth of circuit simulation tools. Circuits are generally designed by iterating several design phases. Before CAD, this consisted of a synthesis stage performed with pencil, paper, and much experience, a capture stage of drafting, and a verification-simulation-analysis stage accomplished with breadboards. Once computers became available, network analysis programs, such as ECAP [7] and NET1 [8], were written for the solution of network equations under steady state or transient conditions. As circuit problems increased in size, additional

tools were needed. Schematic capture aids were constructed to relieve burdensome input to simulation programs, as well as to serve as drafting aids. Circuit simulators were developed for analysis at several model levels, including the transistor, timing, and gate levels. Mixed level simulators now provide mechanisms for dealing with very large circuits [9].

Eventually, circuit simulation tools were integrated with logic level tools from higher up the design hierarchy of Figure 1, so that circuits could be synthesized using various methodologies. Interfaces to tools "below" in the hierarchy allow, for instance, generation of circuit schematics from layouts for design verification. More recently, integrated tools such as Edisim [10] provide well developed user interfaces for interactive analysis of circuits. While much research into various circuit simulation issues continues, circuit simulators have become important parts of current commercial Computer-Aided-Engineering (CAE) systems [11].

In addition to the circuit design tools described above, tools in many other phases of VLSI design have evolved from simple programs to highly sophisticated design aids. For the solution of realistic design problems, a complete design environment is required. Large systems have been developed which integrate CAD tools [12], [13], [14] into such an environment. These systems can be extremely costly; more recently, the trend has been toward developing affordable engineering workstations [2], [15]. These workstations have become very highly integrated, user-friendly tools [16], [17], [18].

Device and process design tools have not yet reached the same degree of development. For the most part, design must be accomplished using a variety of process or device simulators, with only limited interfaces between the user and the programs and between various programs [6]. Few systems both integrate these simulation tools and provide tools comparable to those available in other domains [19]. It is expected that the same evolution followed in other domains of design will occur in process and device design as well. In the next section, the functions important for a complete process and device design environment are examined in more detail.

2 Requirements for a Process and Device Design System

As with each of the other IC design phases, an integrated process and device design system is needed to provide aid in specification, synthesis, capture, verification, simulation, and analysis tasks [20]. As shown in Figure 1, these functions are needed in both process and device design; this chapter will examine each of these needs in turn. The function intended by each task is discussed, and analogies to similar functions in other VLSI design domains are used to pinpoint the intended scope of each function. The capabilities demanded of and by such a tool are examined as well.

2.1 Process Specification

During development of any fabrication process, the engineer is striving to satisfy a large number of goals or requirements. These "specifications" for the process may be formal or informal, existing on paper or only in the engineer's head. Process specification, then, is the task of enumerating the various goals, requirements, and constraints on the fabrication process or fabricated structure.

Various specifications may be needed: thicknesses and compositions of layers, characteristics of two-dimensional topological features (the extent of a bird's beak, for instance), sheet resistances of diffusion regions, junction depth requirements, and so on. To date, facilities for formally specifying the process requirements do not exist; no design automation capability that could use such a process specification has yet been developed. Nevertheless, a process specification function, whether formal or informal, will become an important part of a complete design environment.

2.2 Process Synthesis

Process synthesis serves as the bridge between a process specification and a process description. Given a set of specifications, the engineer or CAD tool must generate a working fabrication process that meets those specifications. This process synthesis function may vary greatly in the degree of automation; two such possibilities are considered below.

2.2.1 Automated Synthesis

In VLSI system design, "silicon compilation" is an almost completely automated synthesis function. Likewise, the ultimate goal of a process synthesis function is to create automatically a fabrication process which meets the process and device specifications. Such "automatic" process synthesis is currently impractical; not only are facilities for process specification completely lacking, but the engineering tradeoffs and dependencies in process and device design are often incompletely understood. Considerable work toward understanding process and device design is required before intelligent tools for process synthesis can be built.

2.2.2 Synthesis by the Designer

A more realistic goal is to provide limited tools which will help the engineer synthesize the design himself. One such tool would be an optimization facility, whereby some measure of the design could be optimized as a function of various process parameters. Tools to help the engineer manage the design as it develops can also be considered "synthesis" aids; design documentation and version management facilities are particularly important.examples of simple "synthesis" tools.

2.3 Process Capture

A means for entering a fabrication process is the process capture function, analogous to schematic capture in circuit design. Process capture requires first of all a "process description," a representation or language for expressing the process [21]. As will be mentioned in secction 6.2, representation of the fabrication process in the face of multiple needs (simulation, documentation, fabrication automation) is a difficult problem. Secondly, a tool is required whereby the engineer can interactively and incrementally enter and edit the process description throughout development.

2.4 Process Verification

Process verification will likely take two forms. First, the process description itself will often need to be checked against various sets of rules. The process description syntax can be validated; syntactic rules such as "you must always specify the time and temperature for a furnace step" can be checked. Furthermore, the process description semantics can be checked, analogous to design rule checking in IC layout. Before simulation, one might verify that a valid device structure can in fact result

from the process [22]. Before recipe generation, the process description can be examined to verify that it satisfies laboratory or fabrication area guidelines.

The second form of the process verification function is to check the finished (or intermediate) simulated structure against the initial process specification. Such a capability would be useful both as part of an automated synthesis loop and as a separate utility available to the engineer.

2.5 Process Simulation

The role of a deterministic process simulator is to model the effects of a fabrication sequence on a wafer. By performing successive simulation steps to model the actual sequence of fabrication operations, one, two, or even three dimensional models of a device or wafer structure can be constructed [23]. Thus, SUPREM-III provides material and impurity concentration information for one-dimensional cross-sections, while SAMPLE [24] calculates two-dimensional geometric effects resulting from lithographic, deposition, and etching process steps.

Process simulators typically manipulate or produce a "wafer structure" or "wafer profile"; this profile may represent not only the impurity concentrations and material composition in the silicon and other layers, but may also represent the topological structure at the surface of the wafer. To date, these representations have been peculiar to each simulator. In the absence of a standard profile interchange format, explicit interfaces between various process and device simulators are required.

In addition to the interface between each simulation program, the interface between the engineer and the process simulator must also be considered. Each simulator currently has its own input language for expressing the process. Creation of these input files (particularly when using multiple simulators or simulating several cross sections of the wafer) is a repetitive and error-prone task. Just as inputs to various circuit simulators can be generated from schematic representations, inputs to various process simulators can be constructed automatically from the process description.

Finally, process simulation typically has large computation requirements. In addition to increased speed in simulation programs, other mechanisms are needed to reduce the amount of time spent performing simulations during process development. For example, a multi-level simulation environment would be useful. In the early stage of a design, simple, computationally inexpensive simulations might be sufficient. Later, more complete and complex simulations can be performed for

more accuracy.

2.6 Process Analysis

A simulated profile, of course, is useful only if facilities are available for the examination and evaluation of that profile. Simple analysis capabilities, such as sheet resistance or junction depth calculations, are often built into process simulators. It is proposed that such analysis functions should not be the duty of the process simulator. Instead, a powerful, general tool for the analysis of process and device information is needed [25]. Such a tool will be feasible once a standard profile interchange format has been developed.

2.7 Device Specification

Just as in process design, there is a need to make explicit the goals and requirements in the characteristics (electrical and structural) of a completed semiconductor device. Because process and device design are tightly coupled, it may well prove that the process and device specification functions are best merged together. In either case, a specification capability is required before any substantial progress toward automated process and device synthesis can be made.

2.8 Device Synthesis

The synthesis function can be viewed as the bridge between a high level expression of the device (the device specification) and an actual representation of the device (as might be constructed by the engineer during device capture). In a larger sense, device synthesis is the global purpose of the device or process engineer as well as of the design environment itself. Stated in this fashion, device synthesis is a huge and unwieldy task. As with process synthesis, the engineer would benefit from several small "synthesis" tools as he undertakes the global synthesis task himself.

2.9 Device Capture

The artifact produced by process design is a "process" or sequence of fabrication steps. The artifact produced by device design, on the other hand, is a model of the device itself. It is at the device capture stage that the interface to process design is made explicit. The device consists necessarily of semiconductor structures; representations of these structures are the results of process simulation. A facility

for capturing this device structure is needed. This function should allow either the direct construction of a device representation, or the "gluing together" of existing profile representations into a complete device representation. As in process simulation, a standard or uniform way of representing the device structure and information pertaining to that structure is strongly needed.

2.10 Device Verification

The verification of a device representation might again take several forms. First, the device representation might be checked before being used in very time consuming device simulations. Secondly, the simulated electrical characteristics might be compared with the device specification to verify that design requirements have been satisfied. Lastly, comparison of simulated devices with measurements from fabricated devices would serve to verify both design and fabrication.

2.11 Device Simulation

Device simulators calculate the electrical characteristics of a given device structure in response to various environmental conditions (i.e., temperature or bias conditions). For example, the threshold voltage of an MOS device or a response to a specified bias condition may be the measure of "success" of a particular fabrication process and device. Analogous to process simulation, the role of the device simulator should be limited to the calculation of these device characteristics.

2.12 Device Analysis

Thorough analysis capabilities are needed in order to evaluate the results of device simulation. It should not be the duty of the device simulator to provide graphical output; rather, a general postprocessor can provide for interactive textual and graphical examination of simulation results. Such an analysis capability can go beyond the simple presentation of results; means for data reduction and manipulation can be provided as well.

The above list of functional needs for process and device design is by no means exhaustive. An environment providing these capabilities, however, would be extremely useful in aiding the design of a baseline device or process. Additional tools to support the realistic design of devices and processes will evolve as they are needed. For instance, the process and device analysis functions will grow to include

tools for variational or yield analysis; the synthesis function will grow to include design centering as well as optimization capabilities. The framework outlined in this section can serve as a guide both to the development of individual process and device design tools and to the integration of these tools into a complete design environment.

3 MASTIF Capabilities

A complete design environment for the development of fabrication processes and semiconductor devices will necessitate growth in current CAD technology. Functional capabilities in specification, synthesis, capture, verification, simulation, and analysis must become available in both process and device design environments. The MASTIF workstation is a first generation attempt at providing some of the functions discussed above. In this section, an overview of MASTIF will be followed by a discussion of the workstation's capabilities. Tools developed as part of this research, as well as tools available elsewhere, have been incorporated into MASTIF; these tools will be described and illustrated in this chapter.

An integrated workstation approach to process and device design has been adopted. The user interacts with a single graphics screen via a tablet and a keyboard; a variety of menus and windows are displayed and available to the user simultaneously. A typical MASTIF screen is shown in Figure 2. Each of the application windows shown in Figure 2 will be discussed in more detail below.

3.1 Process Description Window

A *Process Description Window* provides the engineer with a facility to interactively create and edit a fabrication process (the process capture function). This *Process Description* is independent of any particular process simulator, and is in fact devoid of any simulation directives. That is, the *Process Description* contains only the process as it applies to the entire wafer, rather than a particular simulator or cross sectional "view" of that process. In addition to statements for specifying each process step, the *Process Description* includes constructs for management of various versions and version branches of the *Process Description*. Table 1 lists those statements which are currently part of the MASTIF *Process Description*.

The use of the *Process Description Window* is illustrated in Figure 3. The user enters a process step by typing the name of that step, such as "implant," followed by parameter names and parameter values, such as "arsenic dose = 1e15 energy = 100". The syntax for that step is checked immediately to insure that the step has been entered correctly. The process is displayed in "block mode" in the Figure 3(a), where only the the name for each step is shown. This block mode display is useful for an overview of the process, and shows the different versions that the engineer has tried during development. The same process is also displayed in "sentence mode"

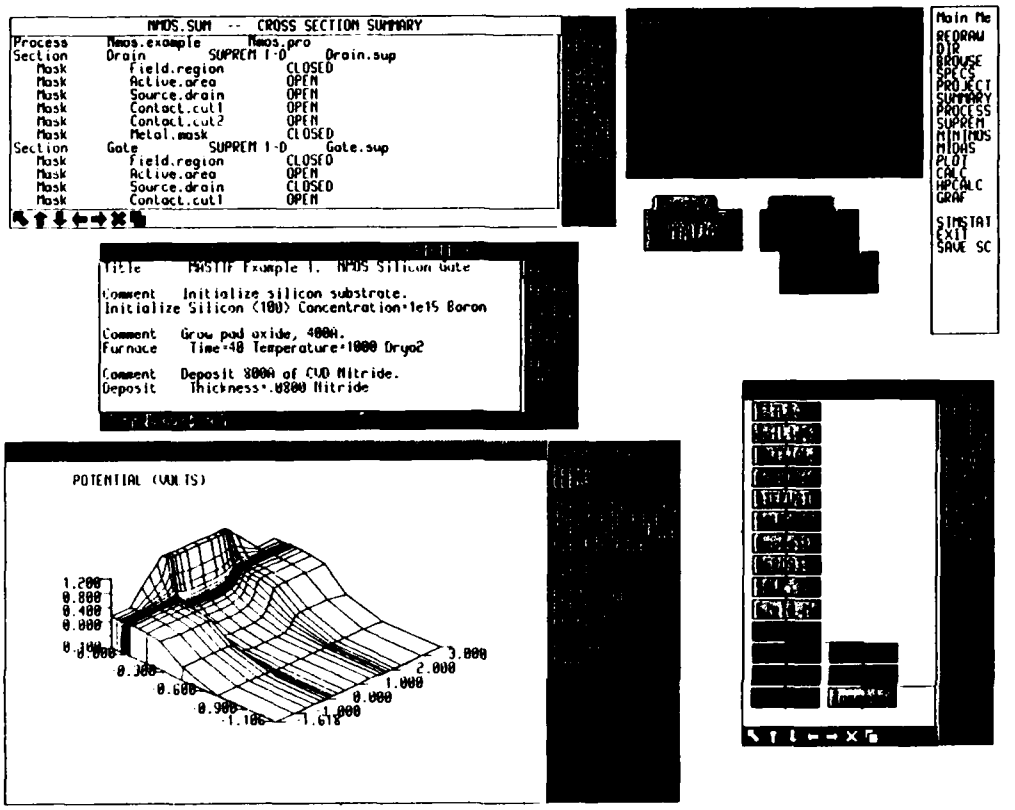
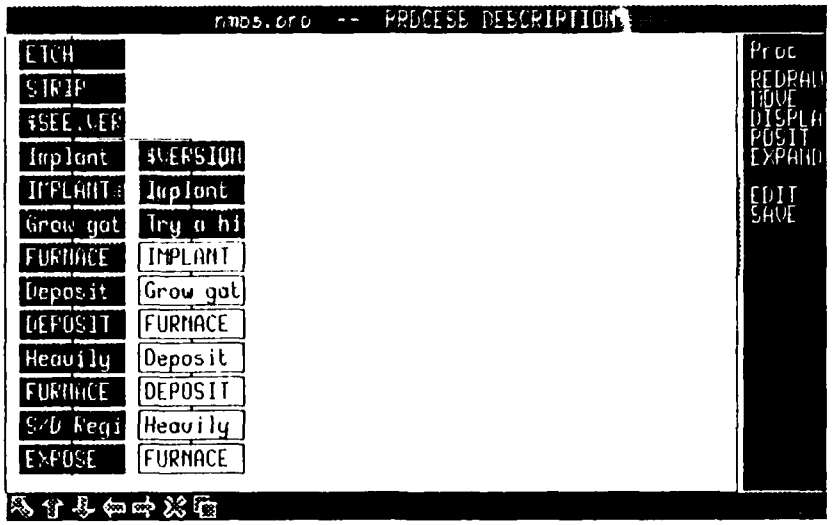
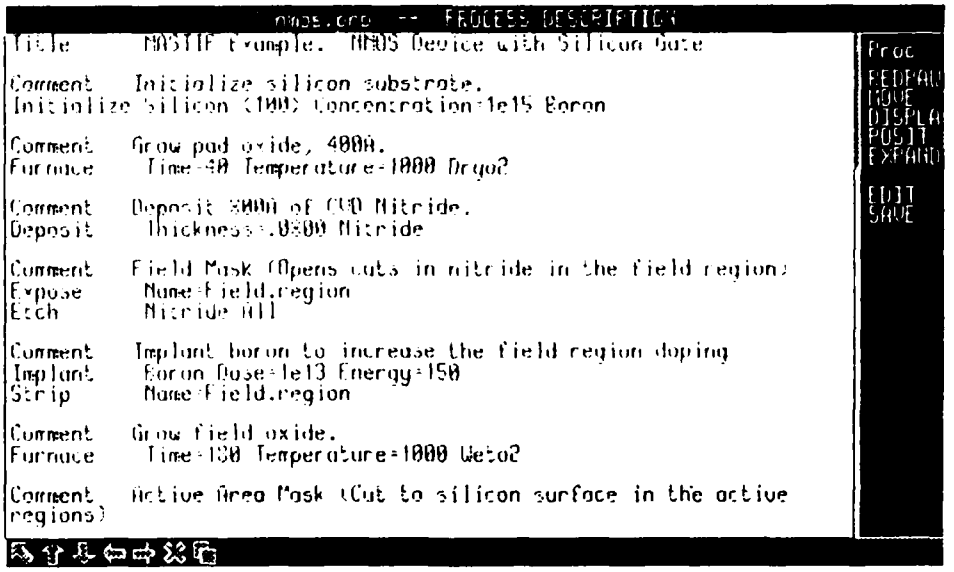


Figure 2: MASTIF screen.
The screen as might be seen for an NMOS project with the MASTIF *Command Area* and *Main Menu* in the upper right corner, the *Cross Section Summary Window* in the upper left, and the *Process Description Window* in the middle left. A *MIDAS Window* in the lower left shows an electrostatic potential surface plot, and a *SUPREM-III Simulation Window* appears in the lower right.



(a) Block display mode. Steps which have been translated into *Process Simulation Windows* are shown in blue, while newly entered steps appear in white. A version branch at the implant step is shown.



(b) Sentence display mode.

Figure 3: MASTIF *Process Description Windows.*

Process Description Statements		
	MASTIF	SUPREM-III
Fabrication Statements	Furnace	Diffusion
	Deposition	Deposition
	Epitaxy	Epitaxy
	Etch	Etch
	Expose	—
	Implant	Implant
	Initialize	Initialize
	Strip	—
Design Documentation Statements	Comment	Comment
	Title	Title
	$See.version	—
	$Version	—

Table 1: MASTIF *Process Description* Statements.
Statements in SUPREM-III are listed for comparison (note that comparable statements are not always available).

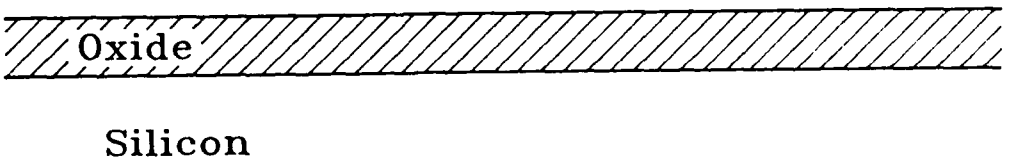
in Figure 3(b), where the whole process step including parameter values is shown.

3.2 Cross Section Summary Window

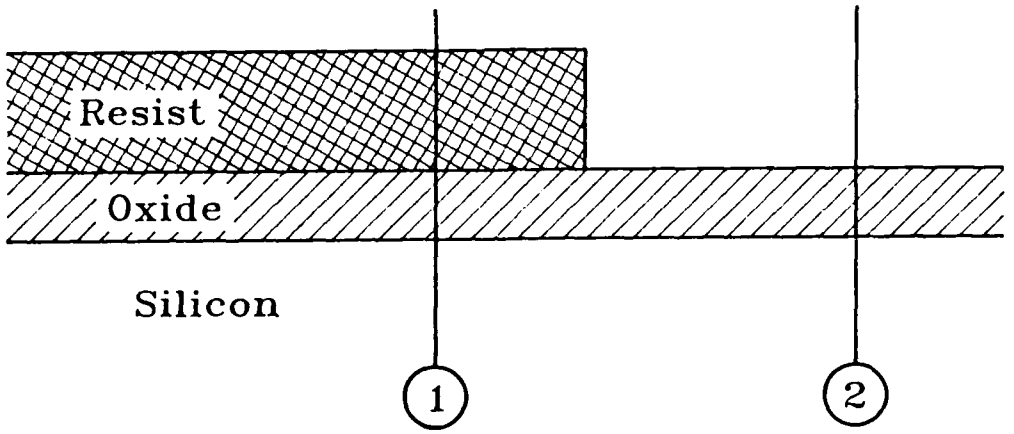
A *Cross Section Summary Window* captures specific mask settings for each distinct cross section, and thus serves as the connection to VLSI layout information. One-dimensional cross sections are currently handled by MASTIF; the differentiation between two cross sections occurs at the "expose" step, as illustrated in Figure 4. Currently, the user must enter the masking information textually. In the *Cross Section Summary Window* of Figure 5, we see the mask definitions for the drain and gate cross sections of a simple nmos process. A *Process Description* and *Simulation Window* corresponding to this *Cross Section Summary Window* are shown in Figures 3 and 6, respectively.

3.3 Process Simulation Windows

The *Process Simulation Windows* provide the bridge between existing process simulators and the MASTIF *Process Description*. Given the overall *Process Description* and specific cross section masking information, a simulation window for a partic-



(a) The profile structure before the *Process Description* line "Expose name=mask.layer1".



(b) Section 1 corresponds to a Cross Section Summary statement "Mask name=mask.layer1 CLOSED"; Section 2 is denoted by a "Mask name=mask.layer1 OPEN" Cross Section Summary statement.

Figure 4: Defining one-dimensional cross sections.

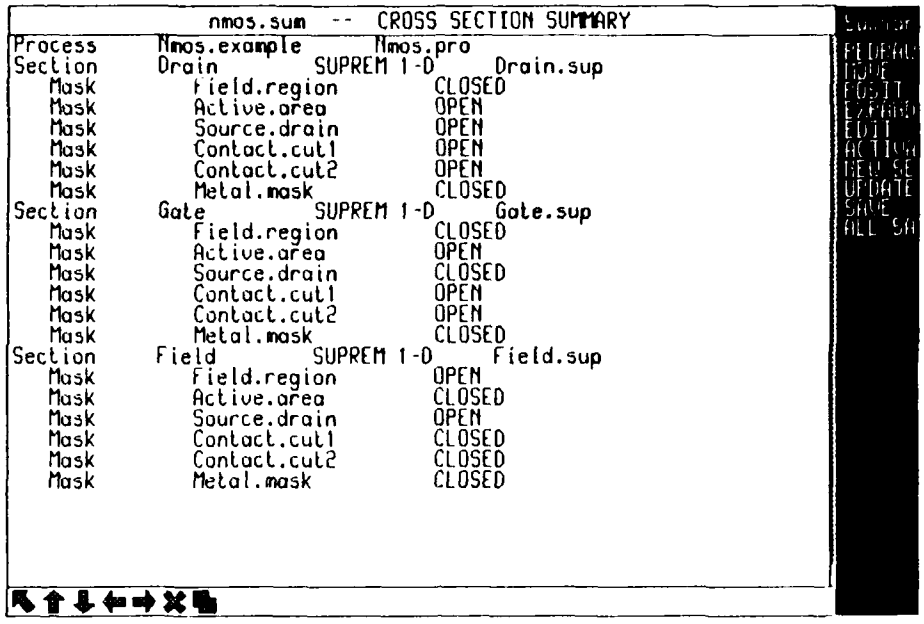


Figure 5: MASTIF *Cross Section Summary Window.* Window shows the process mask settings for each one-dimensional cross section to be simulated.

ular simulator can be created and updated automatically by MASTIF. Currently only *SUPREM-III Simulation Windows* have been implemented; eventually both one and two dimensional simulators will be available.

From the menus of these windows, the user can direct background simulation of particular steps. The color of each step in the *Simulation Window* portrays the simulation status of that step. Red indicates an unsimulated step, yellow that the step is currently simulating in the background, and green that a step has successfully completed simulation. The user can evaluate each simulated step independently by examining its simulated structure interactively.

Figure 6 shows the drain section *Simulation Window* specified in the *Cross Section Summary Window* of Figure 5. The window contains a valid input file to the SUPREM-III simulator. The file depicted in *Process Simulation Windows* is produced by the MASTIF translator rather than by the user; consistent translation into multiple cross sections is thereby assured.

3.4 SUPREM-III Plot Window

The *Plot Window* allows the user to examine the results of a SUPREM-III process simulation graphically (only the results of SUPREM-III simulations can be plotted

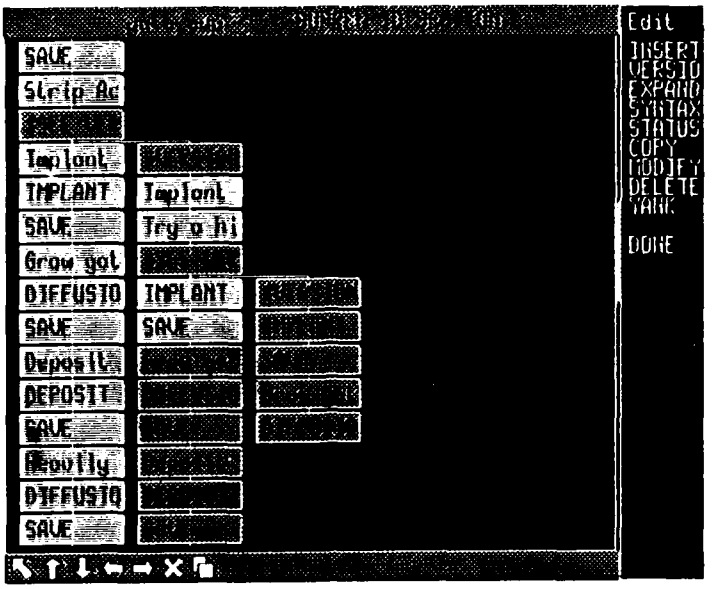


Figure 6: SUPREM-III *Process Simulation Window.*
The input file for the gate cross section is shown here in block mode only.

with this window). The user can interactively change plot parameters without performing any resimulation. In Figure 7, a *Plot Window* corresponding to the last step of the *Process Simulation Window* of Figure 3 is depicted.

3.5 Device Simulation Windows

A *MINIMOS Simulation Window* has been implemented to create and edit input files to the MINIMOS device simulator. While it is possible to run MINIMOS from this window, we have found that there is not a great need for an interactive interface to the simulator, since runs are typically quite time consuming. If MINIMOS runs are sent to a separate computation server, however, such an interface would prove helpful.

3.6 MIDAS Window

The results of device simulation are available for analysis through the *MIDAS Window*, which consists in large part of a MINIMOS postprocessor. A basic feature of the MIDAS system is that it is an interactive tool capable of textual as well as one- and two-dimensional graphical presentation of MINIMOS simulation information.

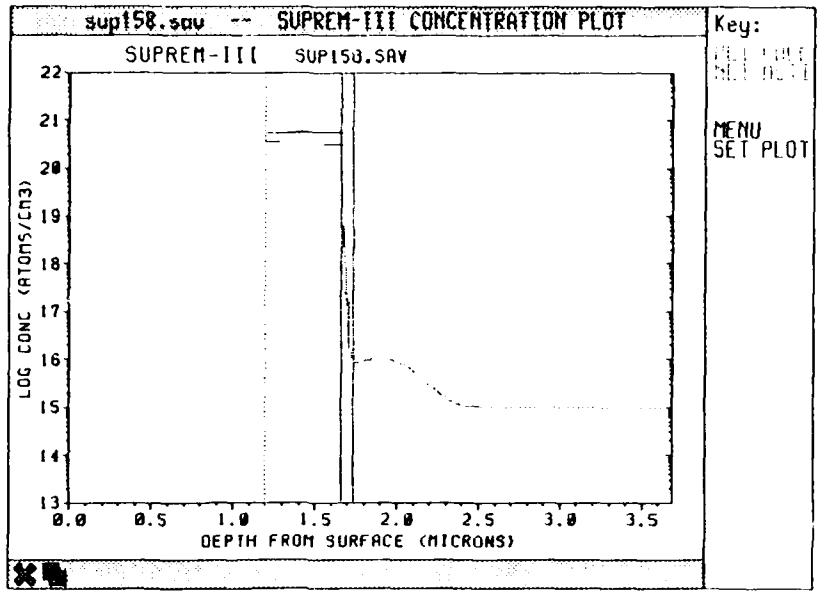


Figure 7: SUPREM-III *Plot Window*
The results of SUPREM-III simulation of an NMOS gate cross section are shown.

3.7 Other Windows

In addition to those described above, a *Browse Window* is available for perusing text files. Through the MASTIF *Specs Window* the user may interactively change the appearance and configuration of the workstation. Additional *Main Menu* functions provide facilities for accessing the underlying operating system, requesting directory listings, saving the state of MASTIF, and examining background process status. Applications currently under development are a general purpose scientific plotting window [26] as well as scratch pad and calculator windows.

4 MASTIF Implementation

This section will summarize the current implementation of the MASTIF workstation. The hardware requirements are first examined, and imbedded software is discussed. Secondly, the overall structure of MASTIF is introduced, with attention paid to the storage of information in MASTIF. Finally, the software modules that have been written as part of this research are examined.

4.1 Hardware and Imbedded Software

The MASTIF "workstation" is composed of several hardware and software components. Our implementation of MASTIF runs on a multi-user VAX 11/750 under the VMS operating system. An AED 767 color graphics display terminal with a digitizing tablet and keyboard completes the hardware for the station. The software consists of

- Individual simulation programs available elsewhere, including SUPREM-III and MINIMOS.
- The MFB graphics package [27]; this package was chosen to achieve some degree of device independence. MASTIF has recently been ported to a Tektronix 4125 display terminal with minimum effort.
- General MASTIF support modules.
- MASTIF application window modules.

The general MASTIF support software is discussed in below. Discussion of the application windows are available in the MASTIF **Users's Manual.**

4.2 MASTIF Information Storage

An important issue in any CAD tool is the choice of data representation. An overriding concern in the implementation of MASTIF was to make the system compatible with existing process and device simulators. As a first step, the station was to serve as a greatly enhanced simulation environment. It was desirable, therefor, that MASTIF be capable of dealing with the standard input files of SUPREM-III and MINIMOS, as well as the binary output files of these simulators. The resulting plan for information storage in MASTIF is illustrated in Figure 8. First, the

Process Description, *Cross Section Summary Window* information, and *Simulation Window* information are all stored as text files. These are human readable, and the *SUPREM-III Simulation Window* files are readable directly by the SUPREM-III simulator. These text files can be understood by MASTIF using the syntax and parsing subsystem described below. The second type of stored information includes the results of simulations. The profile structures and device characteristics are stored using the binary format of the simulators. MASTIF saves the simulated profile structure after each process step, with the philosophy that the increased level of user interaction during process development justifies the additional file storage.

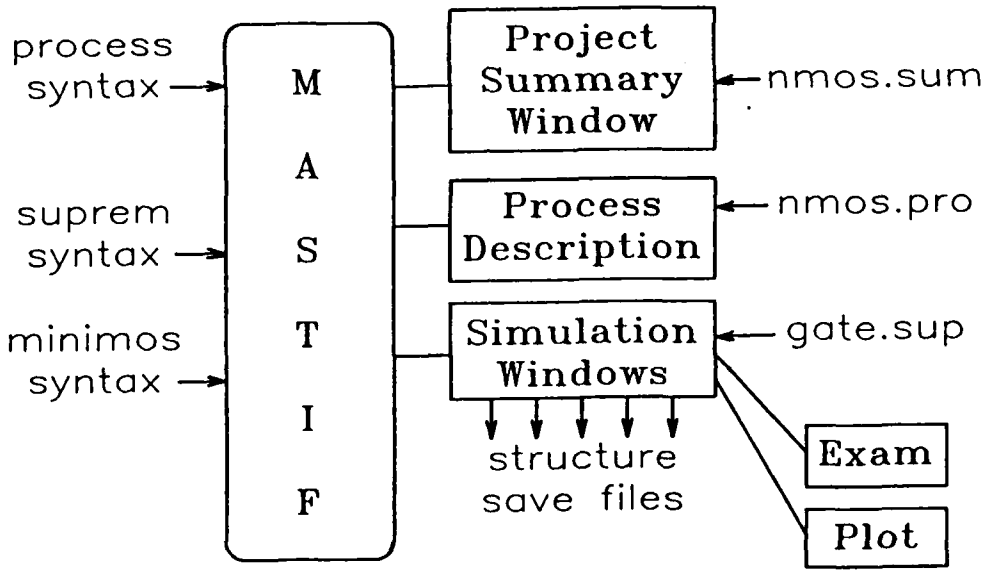


Figure 8: MASTIF information storage.
Window information in MASTIF is stored as text files, while simulated profiles are stored in binary format.

4.3 MASTIF Software

MASTIF currently consists of approximately twenty five thousand lines of C and Fortran (or Ratfor) code. A decision was made to keep MASTIF as transportable as possible. Thus, we have chosen conventional languages (C and Fortran or Ratfor), and have attempted to keep MASTIF independent of various software packages. Other than the MFB package and each simulation program, all MASTIF code is free of imbedded software packages, including the window management subsystem.

The rest of this section will examine each of the various modules written to support the MASTIF application windows.

4.4 Window Management Subsystem

This implementation of MASTIF includes a window manager, consisting itself of a Display Manager, a Menu Handler, and an Input/Output Manager.

4.4.1 MASTIF Windows

The "windows" implemented by MASTIF are all under direct control of the single MASTIF program; each window is NOT controlled by an autonomous (computer) process. To conserve screen display space, a simple window icon facility is included. Typical window functions are performed by the window manager, such as automatic refresh.

4.4.2 MASTIF Menus

Menus in MASTIF can be either permanent or pop-up in nature, and may be textual, iconic, or special purpose (such as a color choice menu). A standard set of icons is provided for window scrolling, deletion, and window display switching. Textual menu options can be chosen either by pointing or by keyboard entry.

4.4.3 Input/Output Handler

MASTIF supports the display and input of text and point information. String prompts may be issued, and textual or point inputs accepted from the keyboard or mouse. Point or string inputs can be filtered through the command processor before being issued to calling procedures, allowing one to execute intermediate commands while in the middle of answering questions to another command. For instance, when confronted with a prompt for a file name, the user may request a directory listing before completing his response.

4.5 Syntax and Parsing Subsystem

Most of the user windows discussed above contain information which may be manipulated interactively by the user. A syntax and parsing subsystem provides a way for the MASTIF programmer to use these same facilities in a new application

window. The programmer must write the "syntax" for application input files; once specified, textual input files may be read by MASTIF, displayed and manipulated by window handlers, and rewritten as text files for storage.

4.5.1 Input File Format

Input files that can be read by the MASTIF parsing subsystem are similar in format to that used by many process and device simulation programs. These input files consist of a sequence of statements; each statement begins with a statement name or label. The statement contains named parameters which may be boolean, character, or numeric in nature. The order of parameters within a statement does not matter; often the case of statements is unimportant as well. A statement can be continued on a following line of text; MASTIF assumes '+' to be the continuation character.

4.5.2 Parameter Specification Grammar

A listing of the statements for a particular application makes up an application "syntax." That is, the syntax is a textual specification of the possible statements, as well as the possible parameters for each statement. In addition, the syntax expresses the logical dependencies of these parameters; mandatory, optional, and mutually exclusive parameters or groups of parameters can be specified. MASTIF provides a simple "parameter specification grammar" that the programmer may use to express the syntax for some application. An example of the syntax statement for a SUPREM-III diffusion line is shown in Figure 9(a). Parameters enclosed in "[ ]" are optional, those enclosed in "( )" are mandatory, and a group of parameters separated by the delimiter "|" indicates that one and only one of that parameter group may be specified in the input file.

Once a syntax file has been written, MASTIF can parse input files for that application. An input line corresponding to the syntax of Figure 9(a) is shown in Figure 9(b). The parameter specification grammar used by MASTIF is capable of expressing the syntaxes of several process and device simulators, and can in general handle input files of the form described above. Syntaxes expressed by MASTIF using this grammar include the *Process Description*, the SUPREM-III input language, the *Cross Section Summary*, and the MINIMOS input language.

```
Diffusion
    Time=<n> Temperature=<n> [T.Rate=<n:0.0>]
    [     (Gas.Concentration=<n> | Solidsolubility)
          (Antimony | Arsenic | Boron | Phosphorus) ]
    [     (DryO2 | WetO2 | Nitrogen)
          [Pressure=<n> ] [P.Rate=<n> ] [HCL%=<n> ] ]
    [ Dtmin=<n> ] [ Dtmax=<n> ]
    [ Dcmin=<n> ] [ Dcmax=<n> ]
    [ Errmin=<n> ] [ Errmax=<n> ]
```

(a) The syntax statement for the SUPREM-III diffusion card [28].

```
Diffusion    temp=1000 time=30 boron solidsol
             dtmin=0.01 dtmax=10.0
```

(b) The input statement corresponding to the syntax line of (a).

Figure 9: Parameter Specification Syntax.
Structure of syntax statements and input file lines understood by MASTIF.

4.5.3 Interactive Input File Manipulator

The interactive input file manipulator provides a menu for creating and editing input files using the above syntax structures. Incremental syntax checking of lines can be performed. These input files can be displayed in either a full textual mode, or in a simple graphic (block mode) format. A standard form editor for the entry of lines which shows the possible choices and parameter dependencies is currently under construction.

4.6 Subprocess Handling Subsystem

A Background/Subprocess Handling Subsystem manages the executions of SUPREM-III and MINIMOS simulations. In order to maintain modularity and extensibility of the system, we have adopted a methodology for inclusion of simulation tools whereby all simulators are maintained in a stand-alone form. MASTIF generates the input files, manages execution, and accesses results of simulators in a manner that does not require modification of the simulation programs themselves. The subprocess manager subsystem thus provides the potential for offline execution of simulators in a networked environment [29].

5 Results

MASTIF is a working system in use at MIT to facilitate process and device design and research. As a first generation attempt at a full process and device design system, it has shown that a highly interactive workstation for process and device design is a valuable tool. On the other hand, a number of drawbacks have been found in the system. It is the intent of this section to examine both the advantages and limitations of the current implementation of MASTIF. Furthermore, this section attempts to condense some of the experience gained in building a design environment for process and device design.

A model of process and device engineering has evolved from this work. The first part of this section will be devoted to this model, and will examine MASTIF in the light of the model. The benefits and disadvantages of the workstation approach used in MASTIF will next be summarized, and comments on the integration of design tools in MASTIF will close the section.

5.1 Model of Process and Device Engineering

An initial goal of MASTIF was that it be well suited to the actual engineering practice of the human designer. Two models of fabrication engineering have evolved during the course of MASTIF development.

The first model of engineering design is "incremental process development," where the engineer constructs an overall fabrication process primarily a single step at a time. A step, or short sequence of steps, is added to the process and the resulting structure is evaluated. Process parameters are modified in the most recently added steps first, in an attempt to meet informal intermediate goals. For example, the engineer may desire a very low resistance drain region. To achieve this, the engineer may establish a goal for a particular intermediate sheet resistance immediately following implantation and activation. The implantation parameters, then, are chosen through knowledgeable trial and error until this goal is approached. By limiting the range of parameter modification, and by allowing fairly loose goals, the incremental development may proceed comparatively rapidly. The result of this engineering effort is a "basic" or structural process. At this point, the second phases of process engineering usually begins.

Incremental development is useful in producing a new process from "scratch." Often, however, a basic process already exists. The task of design is then to modify

the process to satisfy the overall specifications of the process or device. At this stage of development, engineering tradeoffs in process parameter choices must be managed. The "long range" effects of parameter choice must be considered during what is termed here as the "global process development" phase. For instance, changing an implantation energy in an early process step may profoundly affect final profile and electrical characteristics. Both "global"'" and "incremental process development" consist primarily of *modify simulate analyze* loops. The differences are that the range of the loops is larger in global development, and that the loops are typically harder to manage. That is, numerous versions and sub versions tend to evolve, and it becomes a difficult task to keep track of what version is doing what.

An alternate phrasing of these two models is helpful. Incremental development can be thought of as the "structural design" of the device or profile. The basic topology of the fabricated structure is the principal goal. This topology is largely determined by the overall sequence of steps; the exact choice of process parameters is less critical at this stage of process development. The second engineering phase is then directed at "electrical or device design." Process parameters must be modified to achieve the performance goals of the resulting device. The *modify - simulate - analyze* loop broadens in scope, and in practice becomes a *modify process simulate process simulate device analyze device* loop, and can become very time consuming and difficult to manage. As the analysis of the process or the device becomes more thorough (incorporating process sensitivity or yield analyses, for instance), the "global" design task becomes ever more complex.

MASTIF is intended to provide aid in both phases of process and device design. First, MASTIF allows process simulation to be performed on single steps or on short sequences of steps in a highly interactive fashion. Secondly, MASTIF provides an immediate, interactive capability to examine and compare the results of process simulation at any point in the process. And thirdly, MASTIF provides version management capabilities in the *Process Description* and *Process Simulation Windows*. All three capabilities are important in an environment for structural process design. The engineer may add a process step, simulate it, examine the results with a *Plot Window*, and try another version of the step, all in a very interactive fashion.

As pointed out above, device design is an unwieldy task, and MASTIF has less completely developed tools for dealing with the difficulties. In addition to the three capabilities discussed above, MASTIF incorporates an interactive analysis tool

in the *MIDAS Window* to aid in the examination of device simulation results. The interfaces between the engineer and the MINIMOS device simulator as well as between process and device simulators, however, are currently a weak point in the overall system. These interfaces do exist; one may create simulation structures for MINIMOS, though not interactively.

This implementation of MASTIF has attempted to bring together a set of functions to provide help in process and device design. These functions do provide a greatly enhanced environment; the real contribution, however, is that MASTIF provides a framework for the inclusion of additional tools as they are developed.

5.2 Workstation Approach

Given the growing numbers of CAD tools in all aspects of VLSI design, some work has been done elsewhere to provide an environment for the integration and use of these tools. In the simplest cases, these environments consist of assorted simulation programs which the user can run and chain together [30], [31]. On the other extreme, complete operating systems (or even company wide networks) are being developed to provide tools and protocols for the use and integration of CAD programs [32], [33].

The MASTIF project has attempted nothing on the scale of these projects. Instead a single workstation oriented system has been constructed. At the cost of being less complete, we have been able to integrate process and device tools very closely. A second benefit of the workstation approach has been the degree of human-computer interaction made possible. The use of color, quality graphics, and a consistent window oriented user interface are all possible in a workstation environment.

A workstation approach is not without a drawback: workstation hardware is expensive In the typical university or industrial environment every designer may have a terminal, but the availability of color graphics workstations are more limited. Tying software to the workstation (as MASTIF does) requires that the engineer be using the workstation to run individual modules in the MASTIF system. It may be beneficial to have versions of the individual modules that can be executed from either a simple graphics terminal or from a conventional terminal.

The current experimental implementation has a number of additional limitations stemming primarily from its window system. The windowing performed in MASTIF lacks many of the features of full-blown window systems. The MASTIF user cannot edit text files directly using conventional text editors in a screen window. More

importantly, the user can not open an "interaction window" acting as a conventional terminal connection to the machine. While this implementation of MASTIF provides communicating windows for process and device design, the ability to interact with the underlying operating system in familiar ways is lacking.

5.3 Integration of Process and Device Tools

The primary goal of the MASTIF project has been to produce a working tool to aid in process and device design and research. The primary tasks involved in meeting this goal have been to integrate existing tools and to build additional tools. In the course of this project, one limitation has become painfully restrictive. There currently exists no uniform representation for either wafer profile (or structure) information, or device information. As a result, general purpose analysis tools have not been possible, and the interfaces between simulation programs themselves have been difficult. In order to provide additional tools beyond those in the current MASTIF system, a profile interchange format is a must.

Aside from providing individual tools for process and device design, MASTIF has been instrumental in identifying the capabilities required for a complete design environment. Design and implementation of additional tools to fulfill the needs for specification, synthesis, capture, verification, simulation, and analysis can begin in earnest. Even those limited tools currently incorporated in MASTIF allow the engineer to develop a fabrication process at a higher level of abstraction than before possible. Continued development of process and device CAD tools as outlined in Section 2 of this thesis will make effective and timely process design a reality.

6 Future Work

This implementation has been particularly useful in crystallizing thoughts on the functions needed for process and device design. While MASTIF certainly does not provide all of the functions described in section 2, we have made the substantial step of identifying, developing, and incorporating simple capture, verification, simulation, and analysis capabilities that go beyond a limited simulation environment. Specific areas for continued research are mentioned below.

6.1 Profile Interchange Format

The need for interchange formats between related CAD tools has been recognized [34], [35], and a number of interchange formats proposed for various purposes [36], [37], [38]. In process and device design, a need for a common interchange has likewise become painfully obvious; discussion of a "profile interchange format" has begun [39].

A major function of the MASTIF workstation is to provide uniform interfaces between users and various tools as well as between simulators and analysis tools. These interfaces would benefit tremendously from a standard interchange format. The MASTIF workstation has provided useful insight into additional demands on a standard profile interchange format. It is proposed that a powerful, uniform format or representation for profile and device information is critical in a complete design environment, and would fuel the development of functionality (as described in Section 2) that currently does not exist. For example, the segmentation of current simulation programs into separate simulation and analysis tools would be possible. A verification function for comparison of measured and simulated structures is second example.

Work is currently underway to develop a usable interchange format in both ASCII and binary forms [40]. A "general" postprocessing capability built on the format is under investigation, and a prototype is being built for general purpose textual and graphical analysis of information written in the format [26].

6.2 Process Description Language

A representation for the fabrication process is a strong requirement for a process and device design workstation. In this research, the representation has been constructed with a view toward specifying the process for simulation and analysis purposes.

A related issue, however, regards the complete capture of a fabrication process, encompassing not only simulation information, but also recipe instructions, and all other information needed to "completely" define a process [21]. It is expected that the simple process description currently in use will expand to include an interface to the MIT Computer-Aided-Fabrication (CAF) project; given a process description, it should be possible to generate a full recipe for use during fabrication.

References

[1] M. E. Daniel and C. W. Gwyn, "CAD systems for IC design," *IEEE Trans. Computer-Aided Design*, vol. CAD-1, pp. 2–12, Jan. 1982.

[2] P. Agrawal *et al.*, "Workstations: a complete solution to the VLSI designer?," *22nd Design Automation Conference*, pp. 219–225, 1985.

[3] C. P. Ho, J. D. Plummer, S. E. Hansen, and R. W. Dutton, "VLSI process modeling – SUPREM-III," *IEEE Trans. Electron Devices*, vol. ED-30, pp. 1438–1452, Nov. 1983.

[4] D. Chin, M. Kump, and R. W. Dutton, "SUPRA – Stanford University PRocess Analysis program," Tech. Rep., Electronic Research Laboratory, Stanford University, July 1981.

[5] S. Selberherr, A. Schutz, and H. W. Potzl, "MINIMOS – a two-dimensional MOS transistor analyzer," *IEEE Trans. Electron Devices*, vol. ED-27, pp. 1540–1550, Aug. 1980.

[6] K. M. Cham, S. Oh, and J. L. Moll, "Computer-aided design in VLSI device development," *IEEE Journal of Solid State Circuits*, vol. SC-20, pp. 495–500, Apr. 1985.

[7] R. W. Jensen and M. D. Leiberman, *The IBM Circuit Analysis Program*. Englewood Cliffs, NJ: Prentice-Hall, 1968.

[8] A. F. Malmberg, F. L. Cornwell, and F. N. Hofer, "NET1 – network analysis program," Rep. LA–3199, Los Alamos, 1964.

[9] D. E. Thomas Jr. and J. A. Nestor, "Defining and implementing a multilevel design representation with simulation applications," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 135–144, July 1983.

[10] D. D. Hill, "Edisim and Edicap: graphical simulator interfaces," *20th Design Automation Conference*, pp. 608–614, 1983.

[11] *Catalyst Catalogue*. Sun Microsystems, Inc., Mountain View, CA., 1985.

[12] H. Y. Chang, J. D. Pyroe, and R. W. Talmadge, "BELLCAD/EDS an integrated engineering design system," *IEEE International Conf. on CAD, ICCAD-83*, pp. 24–24B, 1983.

[13] S. Nachtsheim, "The Intel design automation system," *21st Design Automation Conference*, pp. 459–465, 1984.

[14] J. C. Foster, "A unified CAD system for electronic design," *21st Design Automation Conference*, pp. 365–369, 1984.

[15] F. K. Richardson, "Important criteria in selecting engineering work stations," *19th Design Automation Conference*, pp. 440–444, 1982.

[16] S. M. Rubin, "An integrated aid for top-down electrical design," *IEEE International Conf. on CAD, ICCAD-83*, pp. 111–112, 1983.

[17] W. H. Kao, M. H. Movahed-Ezazi, and M. L. Sabiers, "ARIES: a workstation based schematic driven system for circuit design," *21st Design Automation Conference*, pp. 301–307, 1984.

[18] G. C. Clark and R. E. Zippel, "SCHEMA an architecture for knowledge based CAD," *IEEE International Conf. on CAD, ICCAD-85*, pp. 50–52, 1985.

[19] A. J. Strojwas, "CMU CAM system," *22nd Design Automation Conference*, pp. 319–325, 1985.

[20] J. S. Mayo, "Design automation lessons of the past, challenges of the future," *20th Design Automation Conference*, pp. 1–2, 1983.

[21] H. L. Ossher and B. K. Reid, "FABLE: a programming language solution to IC process automation problems," Tech. Report 248, Computer Systems Lab., Stanford University, 1985.

[22] P. Kager and A. Strojwas, "PI/C: Process Interpreter/Compiler," *IEEE International Conf. on CAD, ICCAD-85*, pp. 321–323, 1985.

[23] P. Penfield, "Two stage generic process-step model," 1985. Working paper of the CAF project, MIT.

[24] A. R. Neureuther, C. H. Ting, and C. Y. Liu, "Application of line-edge profile simulation to thin-film deposition processes," *IEEE Trans. Electron Devices*, vol. ED-27, pp. 1449–1459, Aug. 1980.

[25] E. J. Farrell, S. E. Laux, P. L. Corson, and E. M. Buturia, "Animation and 3d color display of multiple-variable data: application to semiconductor design," *IBM J. Research and Development*, vol. 29, pp. 302–315, May 1985.

[26] R. B. Duncan, *Implementation of Graphical Analysis and Parsing Utilities for an IC Profile and Device Structure Interchange Format.* Bachelor's thesis, Massachusetts Institute of Technology, May 1986.

[27] G. C. Billingsley, "Program reference for KIC," Memo no. UCB/ERL M83/62, ERL, U.C. Berkeley, Oct. 1983.

[28] C. P. Ho and S. E. Hansen, "SUPREM-III — A Program for Integrated Circuit Process Modeling and Simulation," Tech. Rep. No. SEL83-001, Integrated Circuits Laboratory, Stanford University, July 1983.

[29] S. C. Hughes, D. B. Lewis, and C. J. Rimkus, "A technique for distributed execution of design automation tools," *22nd Design Automation Conference*, pp. 23–30, 1985.

[30] R. Amantea and C. Davis, "MDLGRF, a system of programs for computer-aided modeling of semiconductor devices," *IEEE International Conf. on CAD, ICCAD-83*, pp. 204–206, 1983.

[31] R. J. Skokel and D. B. MacMillen, "Practical integration of process, device, and circuit simulation," *IEEE Trans. Electron Devices*, vol. ED-32, pp. 2110–2116, Oct. 1985.

[32] R. A. Friedenson, J. R. Breiland, and T. J. Thompson, "Designer's Workbench: delivery of CAD tools," *19th Design Automation Conference*, pp. 15–22, 1982.

[33] A. F. Hutchings, R. J. Bonneau, and W. M. Fisher, "Integrated VLSI CAD systems at Digital Equipment Corporation," *22nd Design Automation Conference*, pp. 543–548, 1985.

[34] A. Lowenstein and G. Winter, "Importance of standards," *22nd Design Automation Conference*, pp. 88–93, 1985.

[35] R. J. Pachter, "Computer Aided (CA) tools integration and related standards in a multi-vendor universe," *22nd Design Automation Conference*, pp. 94-95, 1985.

[36] C. H. Parks, "IGES as an interchange format for integrated circuit design," *21st Design Automation Conference*, pp. 273–274, 1984.

[37] *EDIF Specification — Version 1 1 0.* Electronic Design Interchange Format Steering Committee, Nov. 1985.

[38] M. Sugimoto and M. Fukuma, "Standard description form for device characteristics in VLSI's," *IEEE Trans. Computer-Aided Design*, vol. CAD-5, pp. 293–302, Apr. 1986.

[39] A. R. Neureuther, "Profile interchange format," 1985. Personal communications.

[40] D. S. Boning and T. Tung, "A proposed profile interchange format," Apr. 1986. Working paper describing MIT work on PIF and SNC.

END

10-86

DTIC